Beyond Success Rate: A Trajectory Diagnostic Framework for Open Web-Agent Evaluation

Anonymous ACL submission

Abstract

Open web agents are commonly evaluated by task-level success rates in executable browser environments. However, a single scalar score can obscure the assigned denominator, protocol contract, runtime and evaluator failures, and the trajectory evidence behind failure interpretations. We propose a trajectory-level evaluation framework that turns raw browser traces into auditable aggregate claims. At its core are Trajectory Diagnostic Records (TDR), assignment-level records that preserve assigned trajectories, stratify observation/action/evaluator contracts, link diagnostic labels to deterministic evidence, and make aggregate results reversible to individual records. TDR is not a new benchmark, logger, or agent method; it is a paper-facing claim-normalization layer. We instantiate the framework on 2,640 assigned WebArena Lite and WebRL episodes across Llama-3.1-8B, Qwen3-VL-32B-FP8, WebRL-Llama, raw BrowserGym settings, ΔState (a controlled state-interface diagnostic probe), and an official-style VAB WebRL protocol. The audit yields a conservative picture: the strongest BrowserGym seed solves 29/165 tasks, pooled Qwen raw solves 75/495 versus 68/495 for Qwen+ΔState, and WebRL reaches 23/165 only under a separate protocol with 27 evaluator/runtime-confounded rows. Beyond success counts, TDR exposes failure-regime shifts, shortcut-heavy successes, solved-set movement, interface mismatches, and protocol-sensitive interpretations. The result is a reporting discipline: open web-agent papers should state the denominator, protocol contract, runtime/evaluator status, and trajectory evidence behind each aggregate claim.

1 Introduction

Open web agents are increasingly evaluated in executable browser environments, where an agent must observe webpages, issue actions, and complete user-specified tasks through multi-step interaction. Benchmarks and infrastructures such as WebArena, VisualWebArena, WorkArena, OSWorld, and BrowserGym make this setting realistic by providing task goals, browser states, action interfaces, evaluators, and replayable execution traces (Zhou et al., 2024; Koh et al., 2024; Drouin et al., 2024; Xie et al., 2024; Le Sellier De Chezelles et al., 2024). In these environments, a final success or failure is not determined by the model output alone, but by an execution trajectory involving observations, actions, browser-side effects, runtime behavior, and evaluator decisions.

Despite this trajectory-rich process, web-agent papers often summarize performance with task-level success rates. Such scalar scores are convenient for comparison, but they compress away the execution context needed to interpret agent capability: how tasks are assigned and counted, how observations and actions are exposed, how runtime and evaluator failures are handled, and what trajectory evidence supports the final interpretation. This compression is especially problematic for open agents, where reliability is low and similar success rates may arise from different failure sources, protocol assumptions, and degrees of autonomy.

Figure 1 illustrates the claim-unit problem as a contrast between lossy and auditable aggregation. Ordinary reporting funnels assigned trajectories into a single opaque success-rate box, allowing rows to be dropped, protocol contracts to be mixed, and evidence to be omitted. TDR keeps the same assigned rows visible, stratifies BrowserGym and VAB-style contracts, attaches evidence stamps such as loop, contract, shortcut, and artifact pointers, and only then emits auditable claim cards. The figure previews the four principles used throughout the paper: denominator preservation, protocol stratification, evidence-linked diagnostics, and reversible

Code will be released publicly after the anonymity period.

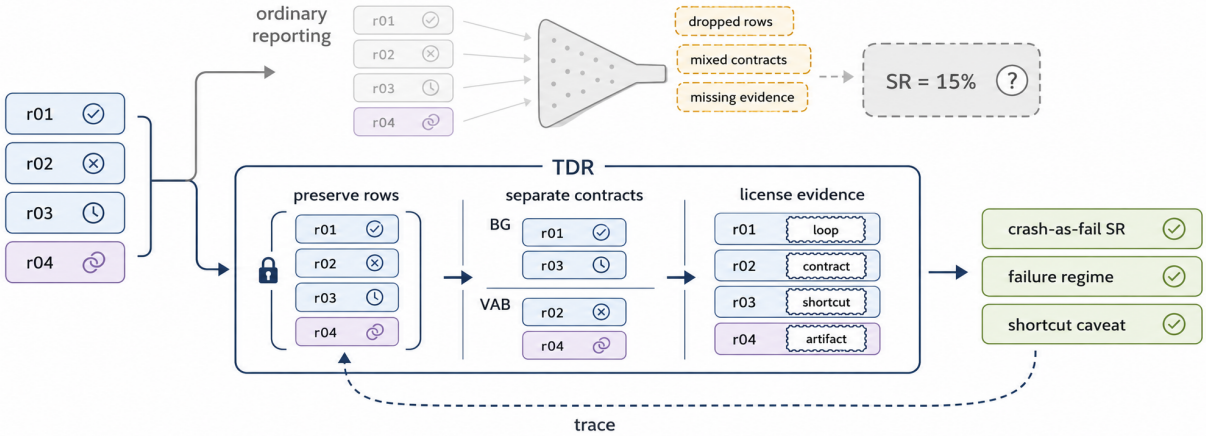

Figure 1: TDR turns assigned trajectories into auditable claims rather than opaque aggregates. Ordinary reporting funnels rows into a single success rate, where assigned rows may be dropped, BrowserGym/VAB contracts mixed, and evidence omitted. TDR preserves denominator status for success, failure, zero-step, and confounded rows; stratifies protocol contracts; licenses loop, contract, shortcut, and artifact evidence; and keeps trace links before aggregation.

aggregation.

We propose a trajectory-level evaluation framework for open web agents that turns raw browser traces into auditable aggregate claims. The framework centers on Trajectory Diagnostic Records (TDR), an assignment-level record abstraction that preserves each assigned trajectory before aggregation. A TDR records the task identity, system and seed, protocol contract, runtime and evaluator status, action evidence, diagnostic predicates, and artifact pointers needed to interpret the final outcome. Based on these records, the framework specifies what information must remain visible when trajectory-level outcomes are summarized into success rates, tables, and figures.

TDR also makes state-interface sensitivity measurable without turning it into a leaderboard claim. Once denominator status, protocol contract, action evidence, and failure evidence are separated, we can ask whether a failure reflects web capability, missing trajectory state, or an interface-level action constraint. We therefore introduce ΔState as a controlled diagnostic state-interface probe inside TDR. For an assigned trajectory, ΔState changes the state information and state-derived action constraints exposed to the agent while preserving task assignment and denominator accounting. Comparing raw and ΔState rows under the same TDR accounting traces changes in success, solved set, shortcut length, and failure labels to record-level evidence rather than to a method-improvement claim.

We instantiate TDR on 2,640 assigned episodes over WebArena Lite and WebRL settings. The study covers prompt-only Llama-3.1-8B, Qwen3-VL-32B-FP8 (Qwen Team, 2025), WebRL-Llama, raw BrowserGym rows, ΔState diagnostic perturbations, and an official-style VAB WebRL row. The evaluation asks four questions aligned with the paper's findings: how reliable the evaluated open agents are under denominator-preserving accounting, whether similar success rates reflect similar failure mechanisms, whether successful episodes demonstrate sustained browser autonomy, and how protocol contracts affect WebRL interpretation.

The answers are conservative. The strongest completed BrowserGym seed solves 29/165 tasks. Across three completed seeds, Qwen raw solves 75/495 tasks, while Qwen+ΔState solves 68/495. WebRL reaches 23/165 only under a separate official-style VAB protocol with 27 evaluator/runtime-confounded rows. Beyond these counts, TDR shows that scalar success can hide short successes, interface mismatches, paired gains and regressions, repeated-action loops, and evaluator-backend sensitivity. Together, these results support a reporting norm rather than a leaderboard claim: success rates should be accompanied by the denominator, protocol contract, runtime/evaluator status, and trajectory evidence that license their interpretation.

Our contributions are:

1. We propose a trajectory-level evaluation framework for open web agents, with TDR as the core record abstraction for preserving denominators, stratifying protocol contracts, linking diagnos-

tics to evidence, and tracing aggregate claims back to trajectories.

2. We provide a reproducible pipeline from assigned trajectories to records, claim-lint reports, tables, figures, and artifact pointers.

3. We use TDR and the ΔState controlled diagnostic probe to audit 2,640 assigned episodes, exposing low open-agent reliability, failure-regime shifts, solved-set movement, shortcut-heavy successes, and WebRL protocol sensitivity.

4. We validate the diagnostic interpretation with a 162-row stratified evidence audit, a local support probe, independent human A/B annotation with adjudication, and WebRL evaluator deconfounding.

2 Related Work and Evaluation Gap

Prior work has improved web-agent evaluation at several layers, but the paper-facing claim unit remains underspecified. Early web-interface and shopping/data benchmarks introduced browser tasks and grounding metrics (Shi et al., 2017; Yao et al., 2022; Deng et al., 2023). Realistic suites define environments, goals, and evaluators (Zhou et al., 2024; Koh et al., 2024; Drouin et al., 2024; Xie et al., 2024). BrowserGym standardizes observation, action, and replay infrastructure (Le Sellier De Chezelles et al., 2024). Method papers compare policies, prompts, training procedures, and interaction interfaces (Qi et al., 2024; He et al., 2024; Lai et al., 2024; Zheng et al., 2024). Broader evaluation efforts motivate multi-metric reporting (Liang et al., 2023; Liu et al., 2024). These layers are necessary, but none specifies which assigned rows entered a paper number, which rows are protocol-compatible, or which trajectory predicate licenses a mechanism statement.

The missing layer is a reporting unit between raw trajectories and scalar results. A benchmark evaluator can say whether a final state satisfies a task, and a replay log can store browser events, but neither specifies which assigned rows remain in the denominator, which runtime or evaluator failures require caveats, which observation and action contracts are comparable, or which trajectory predicate licenses a mechanism claim such as repeated-action loop, shortcut success, or action-interface mismatch. TDR is therefore not another benchmark split, replay backend, or agent policy. It is the normalized record layer a paper uses when it converts trajectories into aggregate claims. As a result, superficially similar success rates can encode different choices about exclusions, protocol compatibility, and evidence strength. The full prior-work comparison is provided in the appendix.

This gap yields four design requirements for paper-facing evaluation. First, denominator preservation requires every assigned trajectory to retain an explicit status, including crashes, zero-step episodes, and evaluator-confounded rows. Second, contract stratification requires observation, action, parser, runtime, and evaluator contracts to be recorded before rows are compared. Third, evidence licensing requires diagnostic labels to be tied to deterministic trajectory predicates rather than inferred from success or failure alone. Fourth, reversible aggregation requires each table entry, figure value, and textual claim to be traceable back to record IDs and artifact paths. TDR implements these requirements in the next section.

3 Trajectory Diagnostic Records (TDR)

TDR operationalizes the reporting gap as an assignment-level record schema. The atomic unit is an assigned trajectory: a task, system, seed, protocol contract, and run status fixed before any aggregation or exclusion. Raw logs and evaluator outputs are inputs to this schema; success rates, protocol comparisons, failure-regime counts, and shortcut statistics are derived views over validated records.

A valid TDR satisfies four invariants. **Denominator preservation** keeps every assigned row visible, including crashes, zero-step episodes, and evaluator-confounded rows. **Contract stratification** records observation, action, parser, runtime, and evaluator contracts before cross-row comparison. **Evidence licensing** ties each diagnostic label to a deterministic trajectory predicate rather than to scalar outcome alone. **Reversible aggregation** requires every reported table value, figure point, and textual claim to recover the record IDs and artifact paths that produced it. Together, these invariants make TDR a claim-normalization layer rather than a logging format.

Two boundaries make this schema useful for review. First, TDR separates record validity from claim strength: a row may remain in the denominator while licensing only a caveat, an unspecified failure, or a protocol-confounded status. Second, TDR separates protocol diagnosis from agent rank-

ing. It can show that two rows are not contract-compatible without discarding either row or merging them into a misleading leaderboard number.

Table 1 translates the invariants into reviewer-facing checks. It asks what remains in the denominator, which rows are contract-compatible, what evidence licenses a mechanism label, whether a success supports sustained autonomy, and whether aggregate claims can be traced back to records.

The construction pipeline is deterministic. It inventories assigned trajectories, normalizes protocol metadata, recovers runtime and evaluator status, canonicalizes actions and model responses, attaches failure and shortcut predicates, and aggregates validated records into tables and figures. The artifact-provided `dstate tdr` subcommands validate records against `diagnostic_record_schema.json`, while paper-facing scripts write aggregate reports and claim-lint tables from paper-local CSV/JSON artifacts.

The same schema fixes how ΔState is interpreted. Within TDR, ΔState is a protocol-controlled diagnostic perturbation: raw and ΔState rows share task assignments and denominator accounting, while state-interface fields and state-derived action constraints are allowed to change. Their comparison is therefore reported through solved-set movement, failure-regime shifts, shortcut changes, and evidence-supported action constraints rather than as a leaderboard method comparison.

The result is a stricter claim grammar. Single-seed gains become diagnostic solved-set movement rather than method improvement. Protocol-mismatched rows are stratified rather than merged. Failure labels require deterministic evidence fields rather than scalar outcomes. Shortcut-heavy success rates are paired with trajectory-length diagnostics before they are interpreted as sustained autonomy.

4 Experimental Setup

This section fixes the benchmark scope, model rows, protocol contracts, and accounting rules before reporting results. Table 2 summarizes which rows are comparable under BrowserGym, which rows are stratified, and what claim unit each row can support.

4.1 Benchmarks and Scope

The main accounting set contains 2,640 assigned WebArena Lite and WebRL episodes: five BrowserGym-style rows run for three seeds on the 165 WebArena Lite tasks, plus one official-style WebRL VAB row on the same 165 tasks. WebArena Lite is the primary benchmark for reliability, failure-regime, shortcut, solved-set, and protocol-sensitivity claims. The VisualWebArena slice checks portability of the TDR workflow only and is not merged into WebArena Lite success-rate claims.

4.2 Models and Systems

The evaluated systems separate prompt-only model capability from diagnostic and protocol effects. Llama-3.1-8B and Qwen3-VL-32B-FP8 provide open-agent baselines. Their ΔState counterparts are matched diagnostic rows: they keep the same task assignments and denominator policy while exposing the controlled state-interface probe described in Sections 1 and 3. WebRL-Llama-3.1-8B is included to test how a trained web-agent policy behaves under BrowserGym-style and official-style VAB contracts.

4.3 Protocol Contracts

Unless stated otherwise, BrowserGym-style rows use raw AXTree observations, the BrowserGym action contract, preserved zero-step and runtime-failed episodes, and crash-as-fail success over assigned rows. ΔState rows keep the BrowserGym action contract but change state-interface fields and state-derived action constraints. WebRL is evaluated under two contracts: a BrowserGym raw-AXTree contract, where interface mismatch is diagnostic, and an official-style VAB contract, which is reported separately because its action setting and fuzzy evaluator are not contract-compatible with the BrowserGym rows.

4.4 Metrics and Diagnostics

The primary metric is crash-as-fail success rate over assigned episodes. We report seed-level and pooled counts, confidence intervals, zero-step counts, short-success counts, solved-set movement, site-wise success patterns, and evaluator-confounded rows separately rather than hiding them in prose.

Diagnostic labels follow the evidence-licensing rule from Section 3. We distinguish repeated-action loops, action-contract violations, execution

Question	TDR rule	Paper-facing claim enabled
What stays in the denominator?	Preserve every assigned trajectory, including zero-step, runtime-failed, and evaluator-confounded rows.	Crash-as-fail rates such as 29/165 and 75/495 are auditable.
Are two rows contract-compatible?	Stratify observation, action, parser, runtime, and evaluator contracts before comparison.	BrowserGym WebRL and VAB WebRL are reported as separate protocol layers.
What evidence supports a mechanism label?	License labels only from deterministic trajectory predicates, not from success or failure alone.	Loop reduction, contract failures, and interface mismatch become evidence-backed claims.
Does a success imply sustained autonomy?	Pair success with trajectory-length and shortcut evidence.	Shortcut-heavy rows such as WebRL BG are reported with $\leq$3-action evidence rather than interpreted as sustained autonomy.
Can aggregate numbers be traced back?	Require every table value, figure point, and textual claim to recover record IDs and artifacts.	Paper figures and tables trace to local CSV/JSON payloads.

Table 1: TDR claim-normalization rules. Columns map each reviewer question to the required record field and the paper-facing claim it licenses.

Row	Model	Episodes	Raw AX	ΔState	BG action	VAB	Claim unit
		BrowserGym WebArena Lite rows					
Llama raw	Llama-3.1-8B	3 x 165	✓	×	✓	×	baseline capability
Llama + ΔState	Llama-3.1-8B	3 x 165	✓	✓	✓	×	diagnostic perturbation
Qwen raw	Qwen3-VL-32B-FP8	3 x 165	✓	×	✓	×	open VLM row
Qwen + ΔState	Qwen3-VL-32B-FP8	3 x 165	✓	✓	✓	×	diagnostic perturbation
WebRL BrowserGym	WebRL-Llama-3.1-8B	3 x 165	✓	×	✓	×	interface mismatch
		Stratified checks					
WebRL VAB	WebRL-Llama-3.1-8B	1 x 165	×	×	×	✓	separate protocol
VWA slice	Qwen3-VL-32B-FP8	30	✓	×	✓	×	portability only

Table 2: Evaluated systems and protocol accounting. Raw AX denotes raw accessibility-tree observations; BG action denotes the BrowserGym action contract; checkmarks mark fields present in each row.

errors, external-navigation prevention, max-step exhaustion, rollout exceptions, evaluator-confounded rows, and unspecified failures. Mechanism claims require deterministic predicates; ambiguous or unspecified rows support denominator accounting only. Quality control combines schema validation, claim-lint checks, a local Qwen3-VL support probe, and blind human A/B validation, with full details in the appendix.

5 Results

Having fixed the benchmark scope, system rows, protocol contracts, and diagnostic metrics in Section 4, we now ask what these records license about open web-agent evaluation. The results are organized as four questions: reliability under denominator-preserving accounting, whether similar success rates hide different mechanisms, how trajectory length constrains autonomy claims, and how protocol contracts change WebRL interpretation. Each subsection answers one question and ends with the corresponding finding.

5.1 How reliable are open web agents under denominator-preserving accounting?

The first measurement question asks whether current open web agents approach reliable WebArena Lite autonomy when every assigned episode remains accountable. We answer this question on the five BrowserGym-style rows because they share the WebArena Lite task set, crash-as-fail denominator, and raw AXTree execution layer described in Section 4. Figure 2 visualizes pooled success rates with uncertainty, while Tables 3 and 4 keep the main-text tabular evidence compact. Full seed-level and site-wise matrices are reported in the appendix.

The capability boundary is low across models, seeds, and sites. The strongest completed BrowserGym seed is Qwen raw seed43 at 29/165, and the strongest pooled row is Qwen raw at 75/495. Site-wise variation is visible, but even the best cells in Table 4 remain far below reliable task completion.

Finding 1: Under crash-as-fail accounting, all evaluated open-agent BrowserGym rows remain far from reliable WebArena Lite autonomy; the

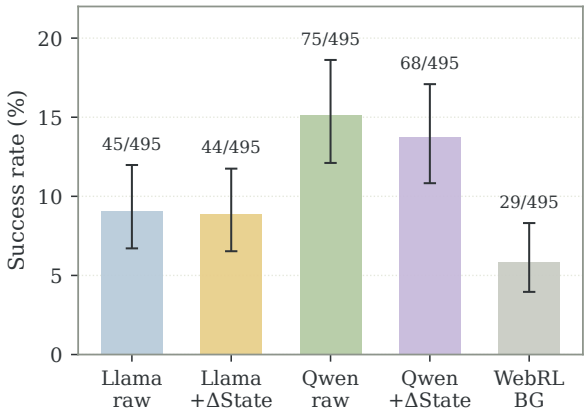

Figure 2: Multi-seed WebArena Lite reliability. The x-axis lists system rows; bars show crash-as-fail pooled success rate, 95% interval, and direct success count.

System	SR	Best	Zero	≤3
Llama raw	9.09%	17/165	23	41/45
Llama + ΔState	8.89%	16/165	28	40/44
Qwen raw	**15.15%**	**29/165**	56	52/75
Qwen + ΔState	13.74%	26/165	84	49/68
WebRL BG	5.86%	13/165	105	29/29

Table 3: Main multi-seed BrowserGym results. SR is pooled crash-as-fail success over assigned episodes; Best is the strongest 165-task seed; Zero counts zero-step episodes; ≤3 counts successes completed within three actions.

Site	Tasks	Best row	Success
GitLab	30	Qwen raw	8/30 (26.7%)
Reddit	19	Qwen + ΔState	6/19 (31.6%)
Shopping	45	Qwen raw/+ΔState	4/45 (8.9%)
Shop admin	35	Qwen raw/+ΔState	8/35 (22.9%)
Map	26	Llama raw/WebRL BG	3/26 (11.5%)

Table 4: Site-wise BrowserGym best cells. Tasks is the number of WebArena Lite tasks per site; Success reports the best or tied row's solved tasks within that site.

strongest single seed solves 29/165 tasks, and the strongest pooled row solves 75/495.

5.2 Do similar success rates imply the same failure mechanisms?

The second question is interpretive: when two rows have equal or similar success rates, do their records support the same failure explanation? TDR answers this question by keeping the assigned denominator visible and comparing evidence fields. ΔState provides the controlled state-interface perturbation for this analysis: it changes the trajectory-derived state summary and state-derived action constraints, but not the assigned task set or denominator policy.

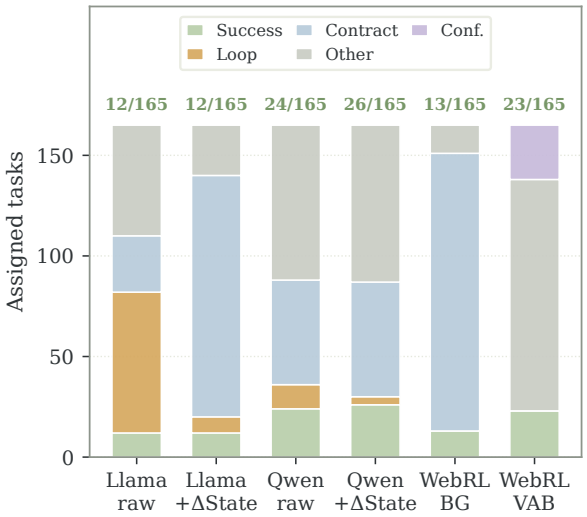

Figure 3: Failure-regime decomposition. Stacked segments are evidence-licensed row counts; BrowserGym rows and the separate WebRL VAB layer expose loop, contract, and confounded regimes.

System	Canon. succ.	Fail label	Fail n	Conf.
Llama raw	12	loop	70	-
Llama + ΔState	12	contract	120	-
Qwen raw	24	contract	52	-
Qwen + ΔState	26	contract	57	-
WebRL BG	13	contract	138	-
WebRL VAB	23	ordinary fail	115	27

Table 5: Canonical failure-regime counts. Canon. succ. is the solved-task count for the canonical 165-task row; Fail label and Fail n report the dominant evidence-licensed failure bucket and its count; VAB ordinary fail denotes valid-scored unsuccessful rows, and Conf. lists evaluator/runtime-confounded VAB rows.

Figure 3 and Table 5 show why scalar equality is not enough. Both canonical Llama rows solve 12/165 tasks, but repeated-action loops fall from 70 to 8 while contract failures rise from 28 to 120. Qwen shows the complementary pattern: the canonical ΔState row moves from 24 to 26 successes, but the three-seed aggregate favors raw Qwen, 75/495 versus 68/495.

Matched solved-set movement clarifies the ΔState claim boundary. At the task-seed level, Llama has 5 ΔState-only successes and 6 raw-only successes over 495 matched pairs. Qwen has 12 ΔState-only successes and 19 raw-only successes, with McNemar p=0.281 for the matched raw versus ΔState comparison. The movement identifies tasks whose outcome changes under a controlled

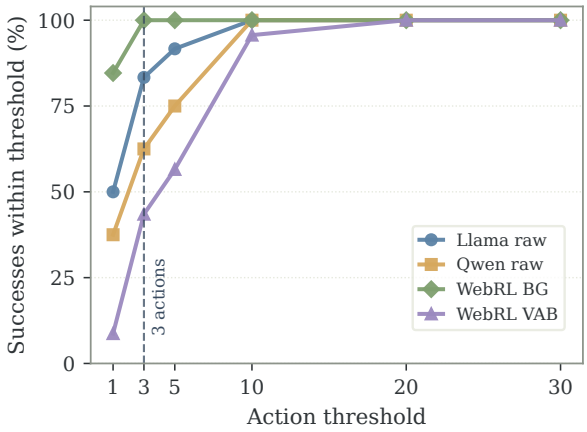

Figure 4: Successful-trajectory length profiles. Curves show cumulative successful episodes by action count; the three-action reference marks short-success concentration.

state-interface perturbation, but it does not support a stable aggregate-improvement claim; the paired movement plot is therefore reported in the appendix.

Finding 2: TDR changes the interpretation of similar scalar outcomes: ΔState shifts failure regimes and matched solved sets, but the evidence supports a controlled state-interface diagnostic rather than an improved agent.

5.3 How much does success rate support sustained browser autonomy?

The third question asks how much a success count licenses claims about sustained browser autonomy. We treat one-to-three-action successes as a trajectory-length diagnostic rather than as invalid outcomes, because some tasks can be solved legitimately in a few actions. The interpretive issue is evidence strength. A row whose successes are concentrated in very short trajectories supports a weaker autonomy claim than a row that succeeds after maintaining progress across search, navigation, forms, and state changes.

Figure 4 shows representative length profiles, and Table 3 reports the corresponding multi-seed short-success counts. Short-success concentration is substantial in the canonical rows: Llama raw and Llama+ΔState each have 10/12 successes within three actions, Qwen raw has 15/24, Qwen+ΔState has 18/26, and BrowserGym WebRL has 13/13. Official-style VAB WebRL is less extreme, with 10/23 successes within three actions and a median successful length of four actions. TDR therefore reports trajectory length beside success, so solved counts are not over-interpreted as sustained multi-step autonomy.

Finding 3: Many successful BrowserGym episodes are one-to-three-action trajectories; success rate should therefore be paired with trajectory-length evidence before being interpreted as sustained browser autonomy.

5.4 How do protocol contracts change WebRL claims?

The fourth question is clearest for WebRL, where a scalar success number is not interpretable until contracts are stratified. We therefore use WebRL as a protocol case study rather than as a reproduction claim. The BrowserGym row evaluates WebRL-Llama through raw AXTree observations and the BrowserGym action parser; the VAB row evaluates the same policy family through an official-style `webrl_id` action contract and a local fuzzy evaluator. These rows answer different accounting questions and should be reported as protocol layers, not merged into one success rate.

Under BrowserGym raw AXTree, WebRL-Llama often emits native `do(...)` and `exit(...)` responses rather than BrowserGym fenced actions. A naive contract-violation rate would overstate semantic action failure. TDR decomposes the canonical BrowserGym seed into 97.37% non-action-text accounting and 2.63% hard violations. The licensed claim is action-interface mismatch under this contract, not semantic infeasibility of nearly all WebRL actions.

Table 6 separates BrowserGym, official-style VAB, frozen-trace, and calibration layers before any WebRL interpretation. The official-style VAB run completes all 165 assigned tasks and obtains 23/165 under crash-as-fail accounting, higher than the BrowserGym canonical WebRL row but still far below reliable autonomy. This count is not evaluator-identical to the WebRL paper score: the VAB fuzzy evaluator's `gpt-4-1106-preview` alias is routed to a local vLLM backend rather than OpenAI GPT-4-turbo, and 27 evaluator/runtime-confounded rows remain in the denominator. Frozen-trace re-judging provides an audit rather than a replacement score, consistent with treating model-judge outputs as sensitivity checks rather than ground truth (Gu et al., 2024). The local judge marks 22/165 correct, and GPT-4.1 agrees with the local judge on 17/20 calibration cases mostly because both judges reject or mark evidence insufficient.

Layer	Assigned	Primary result	Valid-scored	Conf.	Independent check	Claim use
BrowserGym WebRL	3 x 165	29/495 (5.86%)	-	0	canonical 13/165	interface stress test
Official-style VAB	165	23/165 (13.94%)	23/138 (16.67%)	27	local fuzzy scorer	separate protocol
Frozen VAB + local judge	165	22/165 correct	-	-	independent Qwen judge	evaluator audit
GPT-4.1 calibration	20	2/20 correct	-	-	17/20 agreement	sensitivity bound

Table 6: WebRL deconfounding across protocol and evaluator layers. Primary result is the layer-specific assigned-row outcome; Valid-scored excludes confounded scorer/runtime rows where applicable; Conf. counts those confounded rows; Claim use states the licensed interpretation.

Finding 4: WebRL-Llama's interpretation depends on protocol stratification: BrowserGym raw AXTree exposes action-interface mismatch, while official-style VAB reaches 23/165 but remains a separate evaluator-confounded protocol check rather than a reproduction claim.

6 Validation

6.1 Audits

The four findings depend on trajectory-derived evidence, so validation checks whether each claim stays within observed predicates: repeated-action loops, contract/parser/feasibility signals, evaluator/runtime confounds, and short-success evidence.

The six-check package is reported in the appendix: a 162-row deterministic audit with 144 supported and 18 ambiguous rows, a 162/162 support probe, blind human A/B annotation with 0.846 coarse agreement, WebRL frozen-trace judging at 22/165, GPT-4.1 calibration agreement on 17/20 cases, and multi-seed analysis with the best completed seed at 29/165. Together, these checks test trace support, reviewer separability, evaluator sensitivity, and seed stability. They license mechanism claims only where direct evidence exists; ambiguous rows stay unspecified, VAB and BrowserGym rows remain separated, and the VisualWebArena slice is portability-only.

6.2 Claim Boundaries

The main implication is methodological: each success-rate claim should state its assigned denominator, protocol/evaluator contract, runtime/evaluator status, and evidence predicate. TDR packages this requirement as a schema, CLI, checklist, and artifact format. A TDR-backed claim has three fields: row set, contract layer, and evidence license. A count can be a primary reliability result, a diagnostic sensitivity audit, a portability check, or an evaluator audit, but it should not move between these roles without a change in wording. This is why ΔState is reported as state-interface sensitivity rather than an improved agent, why official-style VAB WebRL is a separate protocol check rather than a reproduction score, and why short successes are evidence about claim strength rather than rows to discard.

The role label is part of the claim, not post-hoc metadata. Low success counts locate capability boundaries; matched raw and ΔState movement isolates state-interface sensitivity; BrowserGym and VAB WebRL rows show contract dependence; and short successes calibrate autonomy claims. TDR keeps these interpretations separate before aggregation, so crashes, zero-step episodes, evaluator failures, ambiguous labels, and short trajectories remain visible rather than silently excluded or over-interpreted.

7 Conclusion

Across 2,640 assigned episodes, TDR gives a conservative account of open web-agent evaluation: the evaluated rows remain far from reliable autonomy; ΔState exposes state-interface sensitivity without a stable aggregate gain; and the WebRL and shortcut analyses show why protocol and evidence fields must accompany success rates. The broader lesson is that a web-agent number is a record-level claim, not only a score. With row set, contract layer, and evidence license fixed, the same evidence can support a primary reliability estimate, a diagnostic sensitivity result, a portability check, or a protocol/evaluator audit. Web-agent papers should therefore report the assigned denominator, protocol contract, runtime/evaluator status, and trajectory evidence behind each aggregate claim.

Limitations

Limitations are fourfold: evidence is mainly WebArena Lite, with VisualWebArena only for portability; official-style WebRL uses local vLLM fuzzy matching rather than the paper evaluator; systems cover Llama-8B, WebRL-Llama, and one stronger open VLM with diagnostic variants; and labels are rule-assisted with 162-row blind validation, not full annotation.

References

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. 2024. Workarena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2024. A survey on LLM-as-a-judge. *Preprint*, arXiv:2411.15594.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. Webvoyager: Building an end-to-end web agent with large multimodal models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autowebglm: A large language model-based web navigating agent. *Preprint*, arXiv:2404.03648.

Thibault Le Sellier De Chezelles, Maxime Gasse, Alexandre Drouin, Massimo Caccia, Léo Boisvert, Megh Thakkar, Tom Marty, Rim Assouel, Sahar Omidi Shayegan, Lawrence Keunho Jang, Xing Han Lù, Ori Yoran, Dehan Kong, Frank F. Xu, Siva Reddy, Quentin Cappart, Graham Neubig, Ruslan Salakhutdinov, Nicolas Chapados, and Alexandre Lacoste. 2024. The browsergym ecosystem for web agent research. *Preprint*, arXiv:2412.05467.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, Tianjie Zhang, Wei Xu, Jie Tang, and Yuxiao Dong. 2024. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *Preprint*, arXiv:2411.02337.

Qwen Team. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. 2017. World of bits: An open-domain platform for web-based agents. In *Proceedings of the 34th International Conference on Machine Learning*.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. GPT-4V(ision) is a generalist web agent, if grounded. *Preprint*, arXiv:2401.01614.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*.

A Audit and Validation Summary

Section 6 reports the validation checks used to qualify the mechanism and protocol interpretations in the main text. Table 7 lists each check, its scope, and the boundary it leaves open.

B Prior-Work Reporting Gap

Table 8 expands the compact gap statement in Section 2. It treats TDR as a reporting unit used alongside benchmarks, execution infrastructure, agent methods, and broad evaluation frameworks.

C Taxonomy Audit and Claim Boundary

Table 9 reports the stratified audit behind the coarse failure buckets used in the main paper. The audit asks a narrow question: whether the recorded trajectory fields contain the predicate required by the label. It is not a multi-annotator gold taxonomy. The sample covers the four BrowserGym-style full-165 runs and checks repeated action strings, action-contract counters, execution-error counters, exception strings, evaluator exceptions, and WebRL-specific non-action-text accounting.

For the sampled rows, the audit finds direct evidence for repeated-action loops, ΔState state/action-contract bottlenecks, WebRL BrowserGym contract accounting, execution errors, runtime exceptions, and external-navigation prevention. Overall, 144/162 sampled rows are confirmed. The 18 ambiguous rows come from the unspecified-failure stratum and are kept out of mechanism claims.

As a second check, a local Qwen3-VL support probe was run on the same 162 rows. The probe checked only whether the recorded evidence matched the candidate diagnostic interpretation, not whether the original web task was solved. It marks 162/162 rows as boundary-compatible with the deterministic audit, including 18/18 WebRL contract-accounting rows and 18/18 unspecified-failure rows. The probe is used only as a consistency check over recorded evidence fields.

The human-validation package uses two independent annotations over the blind 162-row sample. Both annotators label the same evidence fields, and disagreements are resolved by evidence-priority adjudication. This validation yields 0.704 primary-label agreement, Cohen's $\kappa = 0.672$, 0.846 coarse-mechanism agreement, coarse $\kappa = 0.812$, 150/162 adjudicated expected-label matches, and 162/162 boundary-compatible adjudications. The 18 rows adjudicated as unspecified or insufficient-evidence match the unspecified-failure stratum. The human validation checks the sampled strata; the full taxonomy remains rule-assisted and tied to recorded predicates.

Table 10 expands the main-text failure-regime counts, while Table 11 summarizes the deterministic, local-support, and human-validation layers.

Table 12 defines the deterministic predicates that license mechanism labels and separates those labels from denominator-only statuses.

D Case Evidence Linked to Aggregate Claims

Table 13 lists representative cases for the aggregate buckets. These examples show the kind of record-level evidence behind each bucket; they are not used to prove that a bucket dominates. Each case comes from the same trajectory records used by the aggregate tables.

E Diagnostic Trajectory Records

Figure 5 and Table 14 give concrete examples of how main-text statements trace back to assigned trajectories. Each displayed row is a paper-audit sample from `paper_manual_taxonomy_audit.md/json/csv`. The compact view shows the preserved row status, the trajectory predicate, and the interpretation supported by that row.

Audit layer	Scope n	Claim checked	Audit result	Claim boundary	Main-text use
Deterministic audit	162	label evidence	144 confirmed	18 ambiguous	mechanism buckets
Qwen support probe	162	evidence support	162/162 compatible	not gold labels	support check
Human A/B	162	label readability	0.846 agreement	adjudicated sample	sanity check
WebRL local judge	165	scorer dependence	22/165 correct	local judge	protocol audit
GPT-4.1 calibration	20	judge sensitivity	17/20 agreement	small sample	sensitivity bound
Multi-seed stability	5 x 3 seeds	seed artifact	best seed 29/165	completed rows	reliability claim

Table 7: Audit and validation summary. Scope n is the checked sample or run count; Audit result reports the positive check; Claim boundary states what the check does not prove.

Work family	What it standardizes	Remaining reporting-unit gap	TDR requirement
WebArena, VisualWebArena, WorkArena	executable browser tasks, goals, and evaluators	assigned rows can still be excluded or merged without a visible denominator rule	preserve every assigned row and report crash-as-fail accounting
BrowserGym and AgentLab-style tooling	observations, action execution, replay metadata, and experiment management	logs do not define which paper-level comparisons are protocol-compatible	record observation, action, parser, runtime, and evaluator contracts
WebRL, WebVoyager, ReAct, SeeAct	policies, prompts, training procedures, and chosen interfaces	model claims can conflate capability with interface mismatch	stratify rows before comparing systems across protocols
HELM and AgentBench-style evaluation	multi-metric reporting and broad benchmark principles	web-agent mechanism claims still need trajectory-level predicates	license failure and shortcut claims through evidence fields
TDR in this paper	assignment-level diagnostic records before aggregation	aggregate claims need an explicit bridge back to assigned trajectories	make every aggregate reversible to record IDs and artifacts

Table 8: Prior-work reporting gap and TDR design requirements. TDR complements benchmarks, tooling, and methods by standardizing assignment-level claim units.

Audit stratum	Population	Sampled	Confirmed	Ambiguous
Low-step successful trajectories	48	18	18	0
Repeated-action loops	90	18	18	0
ΔState state/action-contract bottlenecks	120	18	18	0
WebRL BrowserGym contract accounting	138	18	18	0
Other action-contract violations	80	18	18	0
Action execution errors	46	18	18	0
Runtime or rollout exceptions	44	18	18	0
External-navigation prevention	24	18	18	0
Unspecified failures	62	18	0	18

Table 9: Stratified evidence audit for the trajectory-derived failure taxonomy. Ambiguous rows are intentionally left ambiguous rather than relabeled.

System	Succ.	Loop	Contract	Exec.	Ext.	Rollout	Unspec.	Other	Ord.	Conf.
Llama raw	12	70	28	25	0	7	23	0	-	-
Llama + ΔState	12	8	120	5	0	10	10	0	-	-
Qwen raw	24	12	52	16	12	13	29	7	-	-
Qwen + ΔState	26	4	57	5	4	14	49	6	-	-
WebRL BrowserGym	13	0	138	0	0	14	0	0	-	-
WebRL VAB	23	-	-	-	-	-	-	-	115	27

Table 10: Expanded failure-regime matrix. BrowserGym rows use canonical seed42 counts; WebRL VAB is a separate evaluator-confounded layer.

Validation layer	Scope	Supported	Boundary-compatible	Role
Deterministic evidence predicates	162 sampled rows	144 confirmed, 18 ambiguous	162	Primary audit used for taxonomy claims
Local Qwen3-VL support probe	Same 162 sampled rows	162 supported	162	Auxiliary check over evidence fields, not a gold annotator
Independent human A/B validation	Same blind 162 sampled rows	144 supported, 18 insufficient	162	Human agreement and adjudication over sampled rows

Table 11: Taxonomy validation layers. Support-probe and human checks test evidence consistency while mechanistic claims remain predicate-bound.

Primary label	Evidence fields	Deterministic predicate	Claim boundary
`repeated_action_loop`	normalized actions, target refs, repeat counters	same action family repeats without task progress	supports loop claims only when repeated actions are recorded
`local_navigation_cycle`	URL sequence, page titles, action strings	same local pages or views recur without progress	separated from generic max-step failure
`action_contract_violation`	contract counters, violation reasons, blocked-before-execution flag	parser, target-ref, feasibility, or state-action constraint predicate fires	indicates contract bottleneck, not semantic task impossibility
`webrl_contract_accounting`	WebRL prefix, non-action-text counters, hard-violation counters	native `do(...)` or `exit(...)` text is accounted under BrowserGym parsing	reported separately from hard invalid actions
`action_execution_error`	environment error, Playwright exception, failed action	parsed action reaches environment execution and raises an error	supports execution-layer fragility
`external_navigation_prevented`	blocked URL, navigation guard, exception string	action attempts navigation outside the task environment	supports protocol-safety claims only
`rollout_exception`	zero-step flag, exception class, traceback summary	reset, runtime, launcher, or evaluator exception prevents normal rollout	counted in crash-as-fail denominator
`evaluator_confounded`	evaluator exception, backend route, assertion status	outcome depends on unavailable or substituted evaluator backend	not used as a policy success claim
`max_step_exhaustion`	final step count, success flag, absence of stronger predicate	run reaches step cap without success and without a more specific label	weak mechanistic evidence only
`unspecified_failure`	missing or conflicting evidence fields, ambiguous audit flag	no deterministic predicate above is satisfied	denominator accounting only, not mechanism evidence

Table 12: Deterministic labeling rules for trajectory-derived failures. Mechanistic claims require matching evidence predicates.

Task id(s)	Bucket	Claim supported
95	Qwen capability-threshold case	Stronger open VLM can solve a task where smaller Llama-based agents fail
258, 678	ΔState recovered cases	The controlled state-interface probe can recover individual raw failures without implying a suite-level method gain
139, 155	ΔState lost cases	The same state/action constraints can redirect the policy into new failure modes
173, 211	Qwen capability-threshold cases	Backbone strength changes which tasks are reachable under raw observations
27, 56, 247, 368, 404, 491, 534, 582	Qwen ΔState-only cases	The controlled state-interface perturbation can recover some raw failures without producing a monotonic improvement
23, 43, 93, 95, 173, 205	Qwen raw-only cases	The same state interface can also lose tasks solved by the raw Qwen baseline
4	WebRL interface-mismatch case	Native `do(...)` / `exit(...)` responses are not BrowserGym fenced actions

Table 13: Representative case evidence linked to aggregate claims. Individual cases illustrate claim types but do not establish dominance.

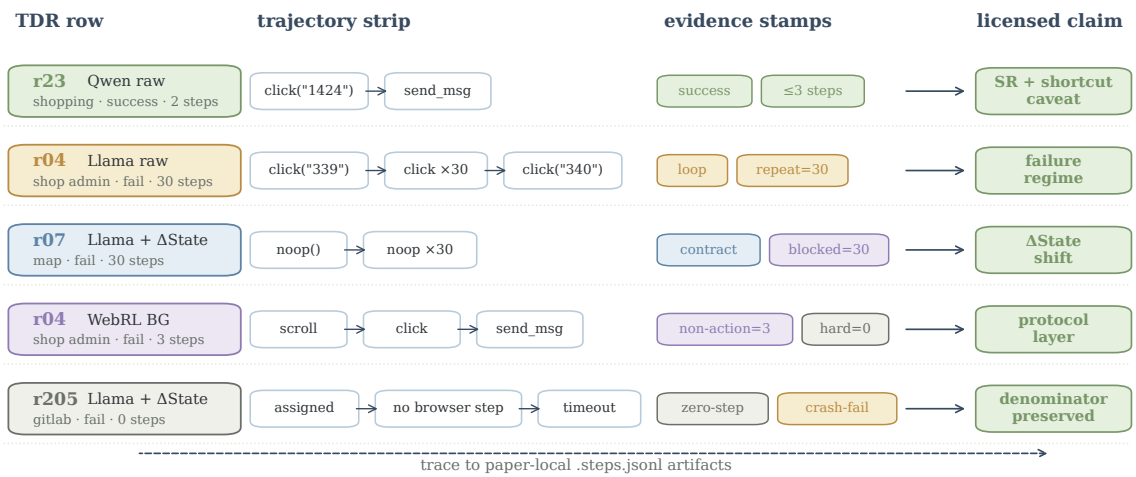

Figure 5: Diagnostic trajectory-record gallery. Each TDR row shows status, evidence, licensed claim, and trace link.

TDR row	Preserved status	Evidence predicate	Claim licensed
Qwen raw task 23, shopping	assigned; success; 2 steps	`click("1424")` followed by `send_msg_to_user`; no contract, block, or execution-error signal	Counts in crash-as-fail SR, but also contributes to the ≤3-step shortcut caveat
Llama raw task 4, shopping-admin	assigned; failed; 30 steps	repeated-action run of length 30 over click actions, with first/last actions `click("339")` and `click("340")`	Licenses a loop failure-regime claim rather than a generic failed-task count
Llama + ΔState task 7, map	assigned; failed; 30 steps	`contract_steps=30`, `blocked_steps=30`, and a repeated `noop()` trajectory under state-action constraints	Licenses the ΔState contract-bottleneck interpretation in Figure 3
WebRL BG task 4, shopping-admin	assigned; failed; 3 steps	BrowserGym-side accounting records `non_action_text=3` and `hard=0` over scroll, click, and final message actions	Separates WebRL BrowserGym contract accounting from semantic policy failure
Llama + ΔState task 205, gitlab	assigned; failed; 0 steps	`Page.goto` timeout occurs before the first browser action; no action strip is available	Preserves zero-step runtime failures in the crash-as-fail denominator

Table 14: Representative TDR rows and licensed claims. The examples compactly render deterministic audit fields used by aggregate tables and figures.

F Completed Supplementary Runs

Tables 15 and 16 report the completed supplementary multi-seed rows. The canonical full-165 rows remain the source for the primary trajectory taxonomy. Additional Llama, Qwen, and BrowserGym WebRL seeds were completed after seed propagation was fixed for both the BrowserGym environment seed and the vLLM sampling seed. These rows are used for low-reliability, failure-regime, and protocol-confound checks; incomplete queued runs are not included in pooled seed claims.

Table 17 and Figure 6 detail the matched Qwen raw/ΔState pairs and the non-monotonic solved-set movement.

G Protocol Details

BrowserGym-style runs use the same 165 WebArena Lite task set and compute crash-as-fail success over all assigned tasks. Raw rows use raw AXTree observations. Rows labeled with ΔState apply the controlled state-interface perturbation and, where stated, full state-action constraints. Zero-step episodes and rollout exceptions remain in the denominator. Valid-scored success is reported only as a secondary diagnostic denominator.

The official-style WebRL run uses VAB WebArena-Lite with `observation_type=webrl`, `action_set_tag=webrl_id`, a 30-step limit, and the released WebRL-Llama checkpoint. It completes all 165 assigned tasks, with 23 successes under crash-as-fail accounting and 23 successes among 138 valid-scored tasks. The remaining 27 tasks are reported as evaluator/runtime-confounded because the fuzzy evaluator's `gpt-4-1106-preview` alias is routed to a local vLLM backend rather than the original OpenAI GPT-4-turbo evaluator. Figure 7 visualizes these protocol-accounting layers.

The artifact also includes a 30-task VisualWebArena portability slice. It uses non-reset reddit and wikipedia/reddit tasks, Qwen3-VL-32B-FP8 raw AXTree observations, the same BrowserGym trajectory logger, the `tdr` build/validate/report commands, and the same failure-review script.

The slice has 30 assigned episodes, 29 valid-scored nonzero-step trajectories, one zero-step `Page.goto` timeout, 1/30 crash-as-fail success, and 1/29 valid-scored success. Its primary failure labels are action-contract violation 8, action-execution error 11, multi-action output 3, repeated-action loop 4, rollout exception 1, success 1, and unspecified failure 2. This slice checks whether the logging, validation, reporting, and failure-review path runs outside WebArena Lite; it is not a VisualWebArena performance estimate.

H WebRL Evaluator-Deconfounding Audit

Table 18 summarizes the WebRL evaluator-deconfounding audit, and Table 19 lists representative GPT-4.1 calibration outcomes. The deconfounding audit freezes the official-style VAB action traces and changes only the evidence scorer, so it measures evaluator sensitivity rather than policy improvement.

The 27 evaluator/runtime-confounded task ids are 5, 7, 11, 18, 19, 30, 35, 48, 51, 54, 57, 81, 85, 89, 91, 104, 109, 114, 117, 118, 119, 120, 121, 129, 140, 144, and 158. Ambiguous cases and state-changing tasks without post-state evidence are counted conservatively. The reported 17/20

System	Seeds	Episodes	Success	Pooled SR	Seed SR mean ± std	95% CP CI
Llama-3.1-8B simple	42,43,44	495	45	9.09%	9.09% ± 1.60%	[6.71%, 11.98%]
Llama-3.1-8B + ΔState v2d	42,43,44	495	44	8.89%	8.89% ± 1.40%	[6.53%, 11.75%]
Qwen3-VL-32B-FP8 raw	42,43,44	495	75	15.15%	15.15% ± 2.19%	[12.11%, 18.62%]
Qwen3-VL-32B-FP8 + ΔState v2d	42,43,44	495	68	13.74%	13.74% ± 1.75%	[10.83%, 17.09%]
WebRL-Llama BrowserGym raw	42,43,44	495	29	5.86%	5.86% ± 1.75%	[3.96%, 8.31%]

Table 15: Completed supplementary multi-seed aggregates. All listed systems now include three completed 165-task seeds under crash-as-fail accounting.

System	Seed	Success	Crash-as-fail SR	Zero-step	≤3-step successes	Main diagnostic
Llama simple	42	12/165	7.27%	6	10/12	repeated-loop primary = 70
Llama simple	43	16/165	9.70%	8	15/16	repeated-loop primary = 70
Llama simple	44	17/165	10.30%	9	16/17	repeated-loop primary = 72
Llama + ΔState v2d	42	12/165	7.27%	9	10/12	contract primary = 120
Llama + ΔState v2d	43	16/165	9.70%	9	15/16	contract primary = 115
Llama + ΔState v2d	44	16/165	9.70%	10	15/16	contract primary = 115
Qwen raw	42	24/165	14.55%	10	15/24	contract primary = 52
Qwen raw	43	29/165	17.58%	14	21/29	contract primary = 45
Qwen raw	44	22/165	13.33%	32	16/22	contract primary = 40
Qwen + ΔState v2d	42	26/165	15.76%	11	18/26	contract primary = 57
Qwen + ΔState v2d	43	21/165	12.73%	36	16/21	contract primary = 50
Qwen + ΔState v2d	44	21/165	12.73%	37	15/21	contract primary = 49
WebRL BrowserGym raw	42	13/165	7.88%	13	13/13	contract primary = 138
WebRL BrowserGym raw	43	8/165	4.85%	46	8/8	contract primary = 111
WebRL BrowserGym raw	44	8/165	4.85%	46	8/8	contract primary = 111

Table 16: Completed supplementary seed-level rows. The strongest BrowserGym seed remains 29/165; WebRL BG seed43/44 successes all finish within three steps.

Qwen setting	Success	Valid-scored	≤3-step successes	Zero-step	Paired relation to raw Qwen
Raw AXTree, seed42	24/165	24/155	15/24	10	reference canonical row
ΔState v2d full constraints, seed42	26/165	26/154	18/26	11	gains 8 raw-failed tasks, loses 6 raw-solved tasks
Raw AXTree, seed43	29/165	29/151	21/29	14	reference supplementary pair
ΔState v2d full constraints, seed43	21/165	21/129	16/21	36	gains 1 raw-failed task, loses 9 raw-solved tasks
Raw AXTree, seed44	22/165	22/133	16/22	32	reference supplementary pair
ΔState v2d full constraints, seed44	21/165	21/128	15/21	37	gains 3 raw-failed tasks, loses 4 raw-solved tasks

Table 17: Completed Qwen matched raw/ΔState pairs. Across seeds 42, 43, and 44, raw Qwen solves 75/495 tasks and Qwen + ΔState solves 68/495, with 12 ΔState-only successes and 19 raw-only successes.

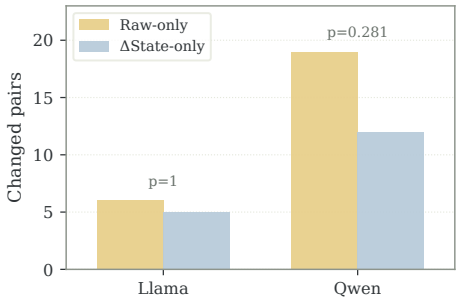

Figure 6: Paired solved-set shifts over three seeds. Raw-only and ΔState-only bars show non-monotonic task movement rather than a stable aggregate gain.

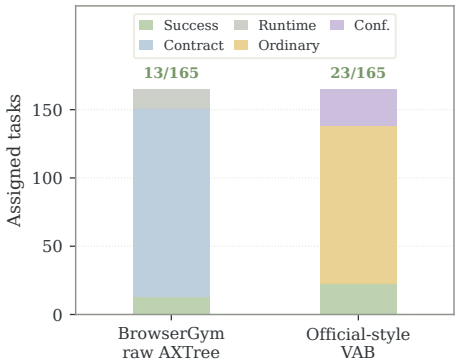

Figure 7: WebRL protocol-accounting layers. BrowserGym exposes action-interface accounting; VAB reaches 23/165 with 27 evaluator/runtime-confounded rows.

GPT-4.1 agreement is dominated by shared incorrect or insufficient-evidence judgments; it should not be interpreted as GPT-4.1 endorsing the 22 local-judge successes.

I Additional Aggregate Evidence

The shortcut analysis reports how many successes finish within three steps. Llama simple and ΔState each solve 12 canonical tasks, with 10/12 such successes. Qwen3-VL-32B-FP8 raw solves 24 canonical tasks, with 15/24 within three steps. Qwen3-VL-32B-FP8 with full ΔState constraints solves 26 canonical tasks, with 18/26 within three steps. Across all three Qwen raw seeds, 52/75 successes finish within three steps; across all three Qwen ΔState seeds, 49/68 successes do so. BrowserGym WebRL solves 13 canonical tasks, and all 13 successes finish within three steps; in seeds 43 and 44, all 16 successes also finish within three steps. The official-style VAB WebRL run is less shortcut-heavy, with 10/23 successes within three actions and median successful length of 4 actions.

Table 20 reports the canonical BrowserGym site counts, and Figure 8 visualizes the same pattern. Qwen3-VL-32B-FP8 is strongest on GitLab and shopping-admin tasks, solving 8/30 and 8/35 respectively, but solves only 1/26 map tasks and 4/45 shopping tasks. BrowserGym WebRL is relatively stronger on reddit and map tasks, but remains low overall. These differences are the reason the main paper reports site-level and trajectory-level diagnostics next to aggregate success rates.

Audit layer	Scope	Result
Original local VAB accounting	165 frozen official-style VAB traces	23/165 = 13.94% crash-as-fail; 23/138 = 16.67% valid-scored; 27 confounded
Confounded-case trace review	27 confounded traces	Four possible success candidates, used only as sensitivity evidence
Independent local Qwen judge	165 frozen traces	22/165 = 13.33%; labels: 22 correct, 96 incorrect, 47 insufficient evidence
GPT-4.1 calibration	Stratified 20-task sample	2 correct, 11 incorrect, 7 insufficient evidence; 17/20 agreement, mostly negative or insufficient-evidence labels

Table 18: Evaluator-deconfounding audit for WebRL VAB. Frozen traces isolate scorer sensitivity.

Task	Site	GPT-4.1 label	Local judge label	Note
54	shopping	correct	incorrect	no Sephora brush reviews found, matching the no-review reference
57	map	correct	incorrect	GPT-4.1 accepts the no-pharmacy-within-20-minutes answer, while the reference lists a pharmacy address
48	shopping-admin	insufficient evidence	incorrect	placeholder date rather than a concrete cancellation date
119	shopping	insufficient evidence	insufficient evidence	final message claims wishlist update, but post-state evidence is missing
121	shopping	insufficient evidence	insufficient evidence	final message claims newsletter subscription, but post-state evidence is missing
129	shopping	insufficient evidence	insufficient evidence	final message claims address update, but billing and shipping DOM evidence is missing

Table 19: Representative GPT-4.1 calibration outcomes. The table lists success candidates and state-evidence-sensitive disagreements from the 20-task sample.

Site	Tasks	Llama raw	Llama + ΔState	Qwen raw	Qwen + ΔState	WebRL raw
GitLab	30	3/30 (10.0%)	4/30 (13.3%)	8/30 (26.7%)	7/30 (23.3%)	3/30 (10.0%)
Reddit	19	2/19 (10.5%)	2/19 (10.5%)	3/19 (15.8%)	6/19 (31.6%)	4/19 (21.1%)
Shopping	45	1/45 (2.2%)	1/45 (2.2%)	4/45 (8.9%)	4/45 (8.9%)	2/45 (4.4%)
Shopping admin	35	3/35 (8.6%)	4/35 (11.4%)	8/35 (22.9%)	8/35 (22.9%)	1/35 (2.9%)
Map	26	3/26 (11.5%)	1/26 (3.8%)	1/26 (3.8%)	1/26 (3.8%)	3/26 (11.5%)

Table 20: Site-wise canonical BrowserGym success counts for primary single-site WebArena Lite groups. Multi-site groups and VAB WebRL are excluded.

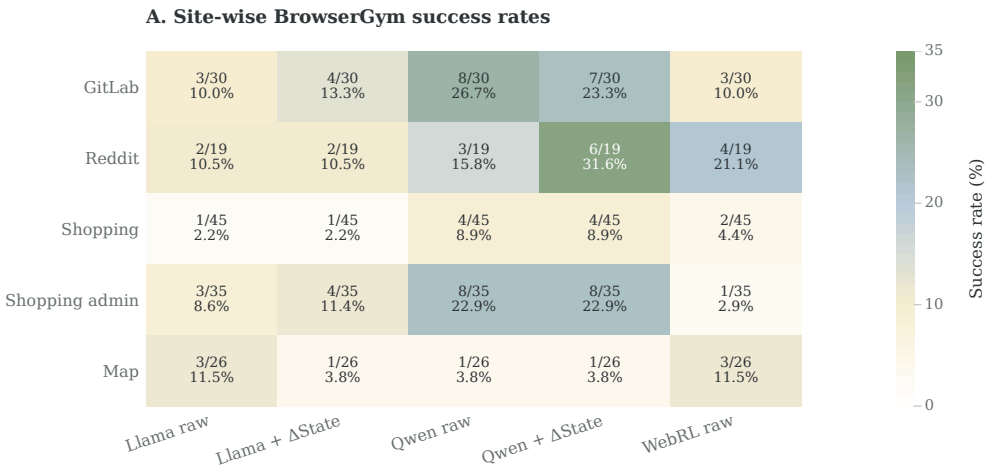

Figure 8: Site-wise success heatmap for primary BrowserGym groups. Strongest cells appear on GitLab, Reddit, and shopping-admin.